

12

DEMANDE DE BREVET D'INVENTION

A1

22 Date de dépôt : 17.01.91.

30 Priorité : 17.01.90 US 466496.

43 Date de la mise à disposition du public de la
demande : 30.08.91 Bulletin 91/35.

56 Liste des documents cités dans le rapport de
recherche : Le rapport de recherche n'a pas été
établi à la date de publication de la demande.

60 Références à d'autres documents nationaux
apparentés :

71 Demandeur(s) : Société dite: SCHLUMBERGER
TECHNOLOGIES, INC. — US.

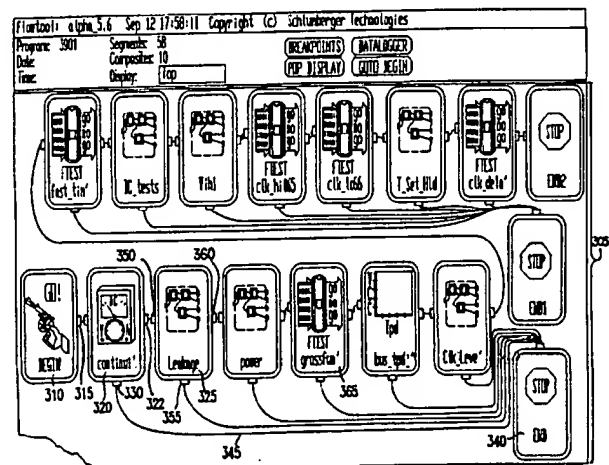
72 Inventeur(s) : Brune William A., Hickling Robert L. et
Poffenberger Russell Elliott.

73 Titulaire(s) :

74 Mandataire : Cabinet Beau de Loménie.

54 Système de commande du déroulement de séquences de tests dans un appareil de traitement de l'informa-
tion.

57 Une interface d'ordinateur est proposée pour un sys-
tème de contrôle de circuits intégrés. l'interface sert à exa-
miner, définir et modifier des paramètres particuliers et des
critères de réussite ou d'échec de tests de fonctionnement,
ainsi qu'à examiner, définir et modifier le déroulement logi-
que des tests de fonctionnement. Chaque test est représen-
té graphiquement par une icône (320, 325 ou 365) sur
un écran d'affichage, et le cheminement logique entre les
tests de fonctionnement est représenté graphiquement par
des lignes (345) connectant des ports d'entrée et de sortie
(330) des icônes. Le branchement logique est graphique-
ment représenté par l'existence de ports de sortie distincts
pour la réussite ou l'échec, pour chaque icône de test, le
port de réussite représentant le trajet logique suivi en cas
de succès au test, et le port d'échec représentant le trajet
logique suivi en cas d'échec au test. Pour modifier le dé-
roulement logique du test, on peut déconnecter et connecter
les lignes d'interconnexion et déplacer les icônes à
l'aide d'un curseur.



La présente invention concerne de façon générale une interface de commande destinée à un système de contrôle de circuits intégrés. Plus spécialement, le mode de réalisation préféré de la présente invention concerne une interface d'ordinateur utilisant
05 des symboles graphiques, qui permet d'examiner et de modifier les paramètres de fonctionnement, les critères de réussite ou d'échec et le déroulement logique de tests de fonctionnement devant être exécutés sur un dispositif à circuit intégré.

Les systèmes de contrôle de circuits intégrés classiques
10 ont fait appel à des paramètres de test et à des séquences de tests possédant un codage imposé. Les paramètres de test et la séquence de tests sont préalablement fixés dans le code de l'ordinateur. Pour changer les paramètres de test et les séquences de tests, il faut donc alors faire sortir, ou éditer, le code de l'ordinateur et
15 avoir connaissance du programme de l'ordinateur et du langage dans lequel le programme a été écrit. Un système connu de ce type est le système de contrôle général S50 (General Purpose Test System) fabriqué et vendu par Digital Test System Group de la société Schlumberger Technologies. Dans ce système, toutes les séquences de
20 tests sont déterminées par un programme d'ordinateur écrit en langage Pascal. Toute modification de la séquence de tests nécessite donc la connaissance du langage Pascal et une reprogrammation du programme de la séquence de tests.

Alors que ce système est souple et peut être utilisé pour
25 contrôler n'importe quel type de dispositif à circuit intégré, son utilisation est limitée aux programmeurs ayant l'expérience du Pascal qui ont été mis au courant des détails du programme de la séquence de tests. Il est donc souhaitable de mettre au point une interface qui permet d'examiner et de modifier de façon générale
30 les tests des circuits intégrés et les séquences de tests sans demander la connaissance des langages d'ordinateur et des programmes.

Dans la série S700 des systèmes de contrôle de cartes
(Board Test Systems), fabriqués et distribués par la société
35 Schlumberger Technologies, les paramètres de test et les séquences de tests sont également programmés en langage d'ordinateur, et il

faut réécrire et recompiler le programme à chaque fois qu'une modification a été apportée à un paramètre de test ou à la séquence de tests. Ce système comporte un éditeur de déroulement de programme de contrôle utilisant des symboles graphiques, qui autorise une
05 définition partielle de la séquence de tests et des conditions de branchement. Toutefois, cet éditeur de déroulement produit simplement le code source en réponse aux demandes introduites par l'utilisateur. De nombreuses instructions, comme par exemple les instructions "If" utilisées pour le branchement conditionnel, doivent
10 encore être définies et insérées par l'utilisateur lui-même.

Un système plus récent, le système de contrôle de mémoires S90 (Memory Test System), fabriqué et vendu par Memory Test System Group de la société Schlumberger Technologies, comporte une interface pilotée par un menu du type texte, qui possède une capacité limitée de modifier les paramètres de test et les séquences de
15 tests sans demander de programmation. Ce système possède une interface pilotée par un menu du type texte qui met en oeuvre un déroulement de programme en "cascade" d'options spécialement conçu pour le classement, par rapport à la rapidité, des produits du type mémoires. Typiquement, un certain nombre de tests réussite/échec
20 sont effectués sur un dispositif de mémorisation à circuit intégré. Si le dispositif échoue à l'un des tests au cours de la séquence primaire de tests, une autre séquence de tests est alors abordée. Typiquement, cette autre séquence de tests correspond à un ensemble
25 d'exigences moins fortes et à un classement en rapidité moins bon. Si la pièce considérée échoue à l'un des tests de l'autre séquence de tests, une série de tests encore moins contraignants, qui correspond à un classement en rapidité encore plus faible, est abordée. De cette manière, chaque pièce est contrôlée et classée selon
30 qu'elle a satisfait à un test d'une série linéaire de tests, ou qu'elle a échoué à toutes les séries. La pièce peut être alors placée dans une "case" correspondant à la série de tests la plus élevée en rapidité qu'elle a réussie.

Toutefois, toutes les séquences de tests du système S90
35 sont disposées dans une succession fixe (linéaire), le déroulement logique qui va d'une séquence ratée à la séquence suivante est

limité à un nombre fixe de points d'entrée, et toutes les modifications de séquences doivent être choisies dans un système de menu du type texte présentant un plan limité. Cette interface limitée ne permet pas l'examen et la modification des séquences de tests

05 d'une manière généralisée. Ainsi, alors que le système S90 est utile pour classer des produits du type mémoires en fonction de leur rapidité, il ne peut pas être utilisé pour le contrôle général d'autres types de circuits intégrés. Par conséquent, il est souhaitable que soit mise au point une interface qui permet d'examiner

10 et de modifier de façon générale des tests de circuits intégrés et des séquences de tests sans demander la connaissance de langages d'ordinateur ou de programmes. En particulier, il est souhaitable que soit produite une interface qui permet d'examiner et de modifier de façon générale le déroulement d'une séquence de tests sans

15 qu'il faille examiner ou écrire le code d'ordinateur.

Selon l'invention, il est proposé une interface d'ordinateur destinée à un système de contrôle de circuits intégrés de type général, qui comporte une interface et des moyens permettant d'examiner, de définir et de modifier des paramètres spécifiques

20 et des critères de réussite ou d'échec relatifs à des catégories particulières de tests de fonctionnement. L'interface offre en outre un langage de programmation spécialisé utilisant des symboles graphiques, qui permet d'examiner, de définir et de modifier le déroulement logique suivi entre les tests de fonctionnement. Chaque

25 test de fonctionnement est graphiquement représenté par un symbole graphique, ou icône, sur un terminal d'affichage et le déroulement, ou cheminement, logique entre les tests de fonctionnement est représenté graphiquement par des lignes reliant les icônes. Le

30 branchement logique entre les tests de fonctionnement est graphiquement représenté par l'existence de ports de sortie, distincts, de réussite et d'échec partant de chaque icône de test. La ligne connectée aux ports de réussite représente le trajet logique suivi si le test de fonctionnement a été réussi, et la ligne reliée au port d'échec représente le trajet logique suivi si le test de

35 fonctionnement a été raté. Une modification du déroulement des tests est prévue en réponse à l'édition de la représentation gra-

phique du déroulement des tests. On effectue l'édition de la représentation graphique du déroulement des tests en utilisant un curseur de traitement graphique pour déplacer les icônes et pour connecter et déconnecter les lignes reliant les icônes. Chaque test de fonctionnement est également représenté, dans le système de contrôle, par des codes objets, ou codes d'exécution de données correspondants, et l'ordre de succession de ces tests de fonctionnement est représenté par des pointeurs connectant d'un point de vue logique les codes d'exécution de données correspondants. Des messages sont envoyés à des programmes, ou routines, appartenant au système de contrôle, à partir des "outils" commandant l'interface d'affichage afin de modifier les codes d'exécution de données de façon qu'ils correspondent aux paramètres des tests de fonctionnement et aux séquences de tests définis par l'interface graphique. Le système de contrôle exécute ensuite les tests de fonctionnement dans l'ordre d'exécution, ou séquence, défini par les codes d'exécution des données.

La description suivante, conçue à titre d'illustration de l'invention, vise à donner une meilleure compréhension de ses caractéristiques et avantages ; elle s'appuie sur les dessins annexés, parmi lesquels :

La figure 1 est une illustration d'un poste de travail connecté de façon à commander un système de contrôle de circuits intégrés ;

La figure 2 est une illustration de l'écran du terminal montrant, sous forme d'affichage graphique, les interfaces du programme "Select" (sélection) et du programme de commande "ControlTool" (outil de commande) ;

La figure 3 est une illustration de l'écran du terminal montrant l'interface du programme "FlowTool" (outil de déroulement), qui montre sous forme graphique des icônes de test et des interconnexions logiques définissant le programme de test sélectionné "3901" ;

La figure 4 est une illustration de l'écran du terminal montrant l'interface du programme "FlowTool", qui illustre sous

forme graphique des icônes de test et des interconnexions logiques définissant l'icône composite du test "Leakage" (fuite) choisi ;

la figure 5 est une illustration de l'écran du terminal montrant l'interface du programme "DCTestTool" (outil de test de courant continu), qui montre sous forme graphique des paramètres spécifiques et des critères de réussite ou d'échec d'un test de courant continu ;

la figure 6 est une illustration de l'écran du terminal correspondant à l'interface "FunctionalTestTool" (outil de test de fonctionnement), qui montre sous forme graphique des paramètres spécifiques et des critères de réussite ou d'échec d'un test de fonctionnement ;

la figure 7 montre l'interface "TimingTool" (outil de synchronisation) résultant de l'ouverture du programme "TimingTool" ;

la figure 8 montre l'interface "LevelsTool" (outil de niveaux) résultant de l'ouverture du programme "LevelsTool" ;

la figure 9 est un affichage "FlowTool" qui illustre graphiquement une séquence de tests existante et un nouveau segment de test, qui n'a pas encore été logiquement relié à la séquence de tests existante ;

la figure 10 est une illustration de l'écran d'affichage correspondant au menu principal et aux menus de segments ;

la figure 11 est un affichage "FlowTool" qui illustre graphiquement une séquence de tests existante et un nouveau segment de test, où un port du nouveau segment de test est logiquement intégré dans la séquence de tests existante ;

la figure 12 est un affichage "FlowTool" qui illustre graphiquement un nouveau segment de test complètement intégré dans la séquence de tests existante ;

la figure 13 est un schéma de principe de l'architecture d'un système de contrôle selon le mode de réalisation préféré de l'invention ; et

la figure 14 est un schéma de principe illustrant les relations qui existent entre les blocs de segments, les définitions de segments et les blocs de données.

Sur la figure 1, est représenté l'environnement du mode de réalisation préféré de la présente invention. En particulier, la figure 1 montre un poste de travail 100 couplé à un clavier 10 et à une "souris" à trois boutons 120. Le poste de travail est en outre
05 couplé à un ordinateur 130 constituant un système de contrôle de circuits intégrés, qui comporte des programmes et des données définissant un certain nombre de tests de fonctionnement devant être exécutés sur un dispositif à circuit intégré 140. Le système de contrôle de circuits intégrés 130 comporte également un équipement
10 de contrôle 135, destiné à la réalisation des tests, constitué par exemple de comparateurs et de dispositifs d'excitation, lequel équipement est couplé au dispositif à circuit intégré 140 afin d'appliquer à celui-ci des signaux et d'en recevoir d'autres en réponse aux programmes et aux données. Le poste de travail 100 com-
15 porte en outre des outils, ou aides, d'interface spécifiques qui coopèrent avec les programmes et les données de l'ordinateur 130 constituant le système de contrôle de circuits intégrés et qui permettent d'examiner et de modifier les tests de fonctionnement et les séquences logiques de tests de fonctionnement, comme décrit de
20 façon détaillée ci-dessous.

Dans le mode de réalisation préféré, le poste de travail 100 est un dispositif d'affichage de symboles graphiques ayant une résolution d'environ un million d'éléments d'image (ou pixels). Le mode de réalisation préféré utilise un poste de travail distribué
25 par la société Sun Microsystems, de Mountain View, Californie, E.U.A. Ce poste de travail et l'ordinateur 130 constituant le système de contrôle de circuits intégrés fonctionnent sous un système d'exploitation Unix et un serveur de traitement graphique à fenêtre X. Les programmes d'ordinateur destinés à l'ordinateur 130 consti-
30 tuant le système de contrôle intégré fonctionnent sous le système d'exploitation Unix et le système de traitement graphique à X fenêtres, qui sont des logiciels appartenant au domaine public, en provenance de M.I.T.

Selon le mode de réalisation préféré de la présente
35 invention, un menu d'outils logiciels est prévu, à partir duquel l'utilisateur du système peut sélectionner un article du menu

appelé "ControlTool" grâce à une manipulation appropriée de la souris 120 ou du clavier 110. L'article de menu "ControlTool" représente un programme de commande du système de contrôle 130, et est décrit de façon plus détaillée ci-dessous.

05

SELECTION D'UN PROGRAMME DE CONTROLE EN VUE DE L'EXAMEN ET DE L'EDITION.

La figure 2 est une illustration de l'écran d'un terminal montrant, sous forme de symboles graphiques, des interfaces du programme "Select" (sélection) et du programme de commande "ControlTool" (outil de commande). La fenêtre "ControlTool" 200 apparaît au bas de l'écran de la figure 2. Lorsqu'on enfonce le bouton droit de la souris 120 alors que le curseur 205 est positionné sur l'appellation 208 "programme" (ce qu'on peut appeler "actionnement du bouton droit sur l'appellation 208", ou "sélection de l'appellation 208"), on obtient l'affichage d'une image-écran du programme "SelectTool" (sélection d'outil) et une interface, comme représenté dans la fenêtre 215. L'image-écran et l'interface "SelectTool" permettent de sélectionner un programme de tests spécifique, comme discuté en détail ci-après.

On se reporte à la fenêtre 215. La ligne 220 de la fenêtre 215 indique que la source des programmes de tests énumérés sous forme de liste est le disque système. L'indication "*" présente sur la ligne "filter" (filtre) 225 indique que tous les programmes de tests sont listés. (Aucun filtre n'est en service). La ligne "path" (trajet) 230 indique le répertoire Unix en cours d'accès.

L'aire de visualisation 235 illustre une liste de fichiers de programmes de tests qui est du type que l'on peut faire défiler. Dans l'exemple représenté, le programme de tests "3901" apparaissant à la ligne 240 a été antérieurement sélectionné. La sélection d'un programme de tests dans la liste de fichiers illustrée dans la fenêtre 235 fait que le nom du programme de tests sélectionné est introduit dans la zone de programme 210 de la fenêtre "ControlTool" 200. Plus spécialement, dans l'exemple représenté, le programme de tests "3901" a été sélectionné et apparaît

dans la zone de programme 210. L'interface permet maintenant l'examen et l'édition du programme de tests "3901", comme discuté ci-dessous.

05 OUVERTURE ET EXAMEN DU DEROULEMENT DES TESTS.

En sélectionnant le bouton de mise au point 250, on obtient l'affichage d'une image-écran "FlowTool" (outil de déroulement) et une interface, comme illustré sur la figure 3, qui permette l'examen et l'édition du déroulement du programme de tests sélectionné, comme discuté ci-dessous.

La figure 3 est une illustration de l'écran de terminal montrant l'interface "FlowTool", qui représente sous forme graphique des symboles, ou icônes, de test et des interconnexions logiques définissant un programme de tests sélectionné. Comme on peut le voir sur la figure 3, les icônes de test et les interconnexions logiques définissant le programme de tests "3901" sont visualisés dans la fenêtre 305. L'ensemble image-écran et interface "FlowTool" sert à établir et à modifier le déroulement d'un programme de tests. Plus spécialement, le déroulement d'un programme de tests est défini par un certain nombre de segments constituant des tests de fonctionnement, représentés sur l'écran par des icônes, et par des séquences logiques des segments de tests, représentées sur l'écran par des ports de réussite ou d'échec et des lignes reliant les icônes.

Le point de départ d'un programme de contrôle, ou programme de tests, est défini par une icône non opérationnelle, c'est-à-dire ne correspondant pas à une opération, illustrée par l'icône "Begin" (commencement) 310. L'icône "Begin" 310 présente un même aspect pour tous les programmes de contrôle de façon à faciliter l'identification du point de départ. Le premier segment de test de fonctionnement du programme de contrôle est identifié par sa connexion à l'icône "Begin" 310 via la ligne d'interconnexion 315. Plus spécialement, une icône "Continuity" (continuité) 320 est connectée à l'icône "Begin" 310 par l'intermédiaire de la ligne d'interconnexion 315.

L'ensemble image-écran et interface correspondant au déroulement du programme de contrôle est en outre défini par certaines conventions d'interface et d'affichage. Si l'on se reporte à l'icône "Begin" 310, à l'icône "Continuity" 320 et à la ligne d'interconnexion 315 de façon plus détaillée, on observe que l'icône "Begin" 310 possède, sur son côté droit, un carré unique appelé port. Ce port indique la circulation normale depuis l'icône "Begin" 310, et l'exécution ou le déroulement du programme de contrôle s'effectue comme indiqué, via la ligne de connexion (verte dans le mode de réalisation préféré) jusqu'à un port situé sur le côté gauche de l'icône "Continuity" 320, qui indique le point de départ de l'icône "Continuity" 320. Dans le mode de réalisation préféré, les ports situés à droite et à gauche des icônes représentant des segments de tests sont bleus, verts ou rouges. Les ports d'entrée sont bleus et sont généralement sur le côté gauche. Les ports de sortie sont généralement sur le côté droit des segments de tests. Il existe deux types de ports de sortie. Un port de sortie vert indique que, si le segment de test associé à cette icône a été satisfait, l'exécution du programme se poursuivra vers le segment de test représenté par l'icône qui est connectée à ce port de sortie. Des ports de sortie rouges, apparaissant généralement au-dessous du port de sortie de réussite, ou sur la partie inférieure de l'icône, indiquent que, si le test associé à cette icône a échoué, l'exécution du programme se poursuivra vers le segment de test représenté par l'icône qui est connectée à ce port de sortie. De plus, on peut définir des ports de sortie multiples dans lesquels la séquence faisant suite à un test de fonctionnement pourra se brancher de manière logique en résultat de la mesure d'une valeur de sortie.

L'icône "Continuity" 320 représente un segment de test de continuité qui possède des critères particuliers de réussite ou d'échec. Si le segment de test de continuité est satisfait, le déroulement du programme s'effectue du port 322 situé sur le côté droit de l'icône "Continuity" à un segment de test représenté par une icône connectée au port 322. Plus spécialement, si le segment de test de continuité a été satisfait, le programme passe à un

segment de test représenté par l'icône de test "Leakage" (fuite) 325. Si le segment de test de continuité n'est pas réussi, le programme passe du port 330 de l'icône "Continuity" 320 à l'icône "Stop" (arrêt) 340. L'icône "Stop" 340 est connectée au port 330 de
05 l'icône "Continuity" 320 par la ligne d'interconnexion 345. L'icône "Stop" 340, tout comme l'icône "Begin" 310, présente un aspect identique pour tous les programmes de contrôle afin de faciliter l'identification du point de terminaison de tous les programmes.

Typiquement, les programmes de contrôle contiennent souvent un plus grand nombre de segments de tests que celui que l'on
10 peut facilement représenter en une seule fois sur un affichage graphique. Afin de pouvoir donner une vue d'ensemble du déroulement d'un programme de contrôle, la présente invention utilise le concept de segments composites et d'icônes composites. Une icône
15 composite est une icône qui représente un certain nombre de segments de tests interconnectés logiquement. L'icône de test "Leakage" 325 est un exemple d'icône composite, qui représente un certain nombre de segments de tests interconnectés logiquement, comme discuté plus complètement ci-après.

20 L'ensemble image-écran et interface "FlowTool" permet d'ouvrir une icône composite, de l'examiner et de la modifier. Le fait de sélectionner l'icône composite du test de fuite 325 permet d'examiner et de modifier les segments de tests et les interconnexions représentées par l'icône 325 du test de fuite. Plus spécialement, en sélectionnant l'icône composite "Leakage" 325, on
25 obtient une image-écran telle que celle représentée sur la figure 4.

La figure 4 est une illustration d'un écran de terminal montrant l'interface du programme "FlowTool" qui illustre graphiquement des icônes de tests et des interconnexions définissant
30 l'icône composite 325 du test de fuite sélectionné. Comme on peut le voir sur la figure 4, l'affichage graphique apparaissant dans la fenêtre 405 montre que l'icône composite "Leakage" 325 est constituée de deux segments de tests interconnectés. Plus spécialement,
35 les ports 350, 355 et 360 de l'icône composite "Leakage" 325,

représentés sur la figure 3, correspondent respectivement aux références 410, 415 et 420 de la figure 4.

05 L'icône de test composite 325 comprend deux segments, le segment de test 425 et le segment de test 430. Dans le cas où les deux segments de tests correspondant aux icônes de test 425 et 430 ont été réussis, le déroulement du programme se poursuit à partir de la référence 410, entre dans le port 435 de l'icône de test 425, satisfait au segment de test correspondant à l'icône de test 425, sort de l'icône de test 425 par le port 440, entre dans l'icône de test 430 via le port 445, satisfait le segment de test représenté par l'icône de test 430, et sort de l'icône de test 430 par le port 450 qui est couplé à la référence 420. Si l'un quelconque des ces segments de tests n'est pas réussi, le déroulement du programme conduit de l'icône représentant le segment de test raté à la référence 415.

15 L'ensemble image-écran et interface "FlowTool" illustré sur la figure 4 comporte également un certain nombre d'autres options, qui sont illustrées dans l'aire de visualisation 455. Par exemple, la sélection du bouton "pop display" (affichage dans le sens remontant) 460 donne lieu à l'affichage de l'image-écran précédente. Dans le présent exemple, ce serait l'affichage du programme de tests "3901", tel que représenté sur la figure 3.

20 Le mode de réalisation préféré fournit également un mécanisme d'interface, désigné sous l'appellation d'icônes "composites". De façon générale, des segments composites fournissent un mécanisme d'interface permettant à l'utilisateur de condenser du point de vue logique la représentation graphique de plusieurs icônes représentant une séquence de tests lorsqu'il le souhaite. Plus spécialement, deux icônes interconnectées, ou plus, peuvent être combinées par application d'un message approprié de l'utilisateur pour être visualisées graphiquement sous la forme d'une icône composite. L'image-écran "FlowTool" résultante ne représentera alors que les lignes d'interconnexion existant entre l'icône composite et les icônes n'appartenant pas à l'icône composite et ne représentera pas les lignes interconnectant les icônes contenues dans l'icône composite. Les icônes et leurs lignes d'intercon-

nexion qui constituent une icône composite peuvent être affichées par "sélection" de l'icône composite. De plus, les éléments d'un segment composite peuvent eux-mêmes être des segments composites, c'est-à-dire que des segments composites peuvent être "emboîtés".

05 Ceci offre beaucoup de souplesse en permettant à l'utilisateur de produire plusieurs vues de la séquence de tests à des niveaux différents, où l'utilisation de segments composites permet à la représentation graphique de rester concise et adaptée aux buts de l'utilisateur au moment de l'examen.

10

MODIFICATION DE DONNEES DE SEGMENTS DE TESTS

Typiquement, le but qui doit être réalisé en premier dans un programme de contrôle est qu'il soit assuré que le système de contrôle est en bon contact électrique avec le dispositif à contrôler.

15 Par conséquent, un test de courant continu du type continuité est souvent le premier test effectué dans un programme de contrôle. On se reporte de nouveau à la figure 3, où l'on voit que l'icône de test 320 représente le cas particulier d'un segment de test de courant continu. Les relations entre ce test de courant

20 continu et les autres tests de fonctionnement du programme de contrôle "3901" sont définies par les images-écran apparaissant sur les figures 3 et 4. Les paramètres particuliers et les critères de réussite ou d'échec de n'importe quel test de courant continu sont

25 établis par une interface de programme "DCTestTool" (outil de test de courant continu) représentée sur la figure 5. Plus particulièrement, les paramètres affichés sur la figure 5 correspondent de façon particulière à un certain segment de test de courant continu, à savoir celui qui correspond à l'icône de test de courant continu 320.

30

L'image-écran de la figure 5 est obtenue par la sélection de l'icône de test 320 de la figure 3. La ligne 505 indique qu'il s'agit de l'affichage du "dcfool" (outil de courant continu), qui produit l'interface de programme "DCTestTool". Le nom du segment de test de courant continu particulier qui est affiché, à savoir

35 "Continuity" (continuité), est indiqué dans la zone "DC" (courant continu) apparaissant à la ligne 510. Ce nom apparaît également sur

l'icône 320 correspondante de la figure 3. (Il apparaît sous forme tronquée en raison de la taille choisie pour les caractères). Les paramètres "TIME" (heure) et "DATE" (date) apparaissant à la ligne 510 indiquent le moment auquel ce test particulier a été modifié pour la dernière fois. La zone "Block Status" (état de bloc) apparaissant à la ligne 515 indique toutes les variations des valeurs courantes du test par rapport à celles emmagasinées sur le disque. Immédiatement au-dessous, se trouve un certain nombre de zones qui offrent les options classiques des tests : "Test Method" (méthode de test), "Measure Device" (dispositif de mesure), "Delay" (retard), "Voltage Connect" (connexion de tension), "Clamp Hi, Lo" (verrouillage haut, bas), "Mask" (masque), et "Result" (résultat). Ces zones permettent la sélection des paramètres de fonctionnement d'un certain nombre d'options classiques associées aux tests de courant continu.

Dans l'aire d'affichage 520 de la figure 5, apparaissent des colonnes intitulées "Pinset" (groupe de broches), "Start" (début), "Stop" (arrêt) et "Ifail" (courant de défaillance). L'aire d'affichage 520 sert à préciser quels groupes de broches prédéfinis du circuit intégré doivent être contrôlés dans ce segment de test. Dans le présent exemple, un groupe de broches comportant toutes les broches ("all pins") a été sélectionné. Par conséquent, cette séquence de tests contrôlera séquentiellement toutes les broches du circuit intégré faisant l'objet du test, sur la base des paramètres suivants.

L'aire d'affichage 530 comporte un dispositif d'interface désigné sous l'appellation d'élément coulissant verticalement, qui sert à afficher et modifier la valeur d'une fonction d'imposition (charge imposée) devant être appliquée au dispositif à circuit intégré qui fait l'objet du test. Celle-ci peut être une tension ou un courant, en fonction de la sélection effectuée à l'aide d'un message d'entrée approprié de l'utilisateur. On définit l'amplitude de la charge appliquée à chaque broche en enfonçant le bouton médian de la souris alors que le curseur est sur l'élément coulissant 535, en déplaçant l'élément coulissant dans le sens vertical par maintien de l'enfoncement du bouton de la souris, et en relâ-

chant le bouton de la souris lorsque l'élément coulissant est au niveau de l'amplitude voulue. L'élément verticalement coulissant affiche aussi l'amplitude exacte à sa surface.

05 Trois éléments verticalement coulissant supplémentaires sont regroupés ensemble. Ils comprennent l'élément "High Limit" (limite supérieure) 540, l'élément "Sense" (détection) 550 et l'élément "Low Limit" (limite inférieure) 560. Les paramètres fixés par l'élément limite supérieure 540 et l'élément limite inférieure 560 établissent le critère de réussite ou d'échec pour ce test de
10 courant continu. Plus spécialement, comme représenté, il n'est imposé aucune limite supérieure. Ceci donne ce que l'on appelle un test "à une seule borne". Une tension limite inférieure de moins un volt (-1 V) est définie par l'élément limite inférieure 560. Ainsi, si la tension sur une broche quelconque est supérieure à
15 moins un volt, le test est réussi. L'élément détection 550 illustre en vert la bande de réussite (-1 à 0 V), et en rouge la bande d'échec (moins de -1 V et plus de 0 V). De plus, des résultats de tests réels, comme représenté en un point d'interruption, peuvent être affichés dans l'élément verticalement coulissant "Sense" 550.

20 L'aire d'affichage "Measure" (mesure) 570 est une liste que l'on peut faire défiler pour sélectionner des broches particulières. Les valeurs réelles du test pour des broches sélectionnées sont affichées dans l'aire d'affichage 550.

L'aire d'affichage 575, qui apparaît dans le coin supérieur droit de la figure 5, affiche des outils particuliers et des
25 données particulières pouvant définir des paramètres de préconditionnement nécessaires pour définir complètement un test de courant continu. Les paramètres de préconditionnement définissent les signaux d'entrée appliqués au circuit intégré soumis au test avant
30 son exécution.

Après que l'examen et la modification du segment de test de courant continu ont été achevés, la sélection de la bande nominale 505 provoque l'affichage d'un menu de descente qui permet la
35 sélection d'une option de fermeture. L'option de fermeture permet la fermeture de l'interface de programme "DCTestTool". La fermeture

de l'outil de test de courant continu ramène l'interface à l'image-écran "FlowTool", dont un exemple a été illustré sur la figure 3.

Un autre segment de test est représenté par l'icône de test "GrossFunction" (fonction brute) 365 de la figure 3. L'ouverture de l'icône de test 365 produit l'affichage de l'outil "FunctionalTest" (test de fonctionnement) qui est représenté sur la figure 6. Les tests de fonctionnement tels que définis par l'outil "FunctionalTest" sont généralement les tâches primaires exécutées par des équipements de contrôle de circuits intégrés à grande vitesse. L'aire d'affichage 605, se trouvant en haut du panneau, de l'outil "FunctionalTest" permet de modifier le nom du test de fonctionnement "grossfun006" affiché dans la zone "Ftest" à la ligne 610.

Les zones d'affichage 615 et 625, situées au bas du panneau de l'ensemble image-écran et interface de l'outil "FunctionalTest", fournissent d'autres zones définissant un test de fonctionnement. Plus spécialement, cette définition de test se présente sous une forme hautement modularisée et combine des définitions et des paramètres à partir de données incluses dans les blocs de données définis par diverses zones. Par exemple, des paramètres de synchronisation utilisés pour l'application de stimuli d'entrée dans le dispositif et servant à mesurer les signaux de sortie du dispositif faisant l'objet du test sont définis dans le bloc de données identifié dans la zone appelée "Timing" (synchronisation) à la ligne 620. La zone "Timing", la zone "Levels" (niveaux), la zone "Code" (code) et les zones "Open Pins" (broches en circuit ouvert) sont actives et on peut les "ouvrir" pour produire l'affichage graphique d'un outil particulier qui permettra de visualiser et de modifier le bloc de données définissant cet aspect particulier du test. De la même façon, une liste de configurations peut être spécifiée à l'aide de la zone "Vector" (vecteur) à la ligne 635. Des points de début et d'arrêt peuvent être définis si toute la configuration ne doit pas être exécutée, et un masque peut être défini si certaines valeurs ne doivent pas être échantillonnées. Les configurations, qui sont des séquences binaires relatives à chaque broche

d'entrée et de sortie, sont définies par un outil spécialement prévu pour cette fonction, l'outil "Vector" (vecteur).

La figure 7 montre l'interface "TimingTool" (outil de synchronisation) qui résulte de l'ouverture du programme
05 "TimingTool". Comme indiqué, les données correspondent au code d'exécution de données "grossfuncTIM006" indiqué dans la zone "Timing" à la ligne 620 de la figure 6. Le panneau supérieur 710 contient une zone "Timing" à la ligne 715, qui définit le code d'exécution de données correspondant au présent affichage. La ligne
10 716 comporte une zone "PINDEF TABLE" (table de définitions de broches), qui énumère les codes d'exécution de données définissant les broches du dispositif sélectionnées en vue d'être soumises au test.

L'aire d'affichage 720 illustre de façon générale un
15 diagramme du type oscilloscope définissant diverses formes d'ondes de signaux. Avec cet outil, on peut modifier la synchronisation des transitions de niveau en "traînant" les flancs des formes d'ondes d'une position à une autre. Plus spécialement, les transitions, par exemple la transition d'un niveau bas à un niveau haut 725, corres-
20 pondent à une aire active contenant l'affichage de la transition. En enfonçant le bouton médian de la souris alors que le curseur est dans cette aire active, on peut "traîner" horizontalement la transition jusqu'à une nouvelle position correspondant à un instant nouveau, et la repositionner à cet instant nouveau en relâchant le
25 bouton de la souris.

Dans la partie droite de l'aire d'affichage 730, se trouvent deux boutons-poussoirs 735 et 740. Le fait de sélectionner le bouton-poussoir "TEST" 735 charge immédiatement les conditions courantes définies par l'interface "TimingTool" dans l'équipement
30 matériel exécutant le test, fait effectuer le test et ramène un résultat de réussite ou d'échec. Ceci permet de contrôler rapidement un dispositif à l'aide de nouveaux paramètres de test. Par sélection du bouton-poussoir "CHECK" (vérification) 740, on vérifie l'existence de violations de contraintes de l'équipement de test.

35 La colonne gauche 745 de l'aire d'affichage 720, intitulée "PINDEF" (définition des broches), contient des broches dis-

05 tinctes ou des groupes de broches (appelés "pinsets") du dispositif à circuit intégré faisant l'objet du test. Par exemple, les premiers articles "a" et "b" de la colonne 745 représentent chacun des bus à quatre bits définis par l'interface "PinDef" (définition des broches).

10 La colonne 750 est appelée "SEQUENCE NAME" (nom de séquence). La colonne 750 des noms de séquences contient les noms attribués par l'utilisateur pour définir la forme d'onde associée affichée horizontalement à droite dans l'aire d'affichage "WAVEFORM DESCRIPTION" (description des formes d'ondes) 755.

15 Des éléments de synchronisation et de présentation précisés dans l'interface "TimingTool" définissent donc, lorsqu'ils sont combinés avec des séquences de "1" et de "0" et des niveaux définis par d'autres outils, les formes d'ondes devant être appliquées ou soumises au test. Le fait que ces définitions soient séparées permet de les faire varier indépendamment. Ceci permet d'utiliser un groupe de paramètres de synchronisation sur plusieurs groupes différents de niveaux, ou d'utiliser un groupe de niveaux avec plusieurs groupes différents de synchronisations.

20 En ce qui concerne l'article de "PINDEF" intitulé "y" dans la colonne 745, on note que deux formes d'ondes différentes lui sont associées. La forme d'onde de la ligne 760 illustre un échantillon 765. Un échantillon correspond au moment d'un test effectué sur la broche "y" lorsque le signal de sortie du dispositif est échantillonné et comparé suivant un critère particulier de réussite ou d'échec. Dans le mode de réalisation préféré, l'échantillon 765 est visualisé sous la forme d'une petite région ayant du vert à la partie supérieure et à la partie inférieure et du rouge à la partie moyenne pour indiquer qu'elle représente une combinaison particulière de "1" et de "0" formant un signal de sortie sur la broche "y". Plus spécialement, le critère de réussite est haut ou bas, et non pas entre les deux amplitudes définies.

35 Dans l'image-écran "TimingTool", la synchronisation est entièrement définie sous forme de configurations de "0" et de "1" définies par un outil de configuration. La synchronisation particulière de chaque transition "0" ou "1" est définie par rapport à

"T0". "T0" est le début d'un vecteur de test. Ainsi, le "T0" suivant est la fin du premier vecteur de test et le début du vecteur de test suivant. Généralement, en appliquant un groupe de 1-0 à une broche et en contrôlant, à un certain instant ultérieur, le signal de cette broche, ou d'une autre broche, on définit un "vecteur test".

On ouvre le programme "LevelsTool" de la même manière que le programme "TimingTool". Plus spécialement, le fait d'activer le bouton gauche alors que le curseur est sur la zone "Levels=" (niveaux=), à la ligne 130 de la figure 6, ouvre le programme "LevelsTool". La figure 8 illustre l'ensemble image-écran et interface "LevelsTool", que l'on obtient en ouvrant le programme "LevelsTool".

Comme on peut le voir sur la figure 8, une zone "Level Block" (bloc de niveaux) apparaissant à la ligne 105 indique le code particulier d'exécution de données qui doit être examiné. Une aire d'affichage 820, que l'on peut faire défiler, affiche les groupes de broches qui ont des niveaux définis. Le fait de sélectionner un groupe de broches sur l'aire d'affichage 820 provoque l'affichage graphique des paramètres de niveau dans l'aire d'affichage 830. Un groupe d'éléments verticalement coulissants 840, 850, 860, 870 et 880 permet de définir les amplitudes prises pour VIH (tension d'entrée haute), VIL (tension d'entrée basse), VOH (tension de sortie haute), VOL (tension de sortie basse), et Vref (tension de référence). Les titres des colonnes sont "Pinsets" (groupes de broches), puis VIH, VIL, VOH, VOL et Vref.

L'aire d'affichage 890 est constituée de deux éléments verticalement coulissants qui définissent la charge programmable. IOH (courant de sortie haut) et IOL (courant de sortie bas) sont les paramètres du courant.

Si l'on revient à l'image-écran "TimingTool", représentée sur la figure 7, on note que les divers éléments de "PINDEF" sont définis en termes de "0" et "1". Les valeurs quantitatives de ces tensions ne sont pas précisées par le programme "TimingTool". Au contraire, les niveaux réels de tension de ces broches d'entrée sont définis par le programme "LevelsTool". A tout instant, chaque

broche d'entrée peut être excitée à un niveau haut ou bas. Ainsi, chaque broche d'entrée définie recevra un bit de donnée fonctionnelle de la part du vecteur de test. Un dispositif d'excitation, au sens matériel, de l'appareil de contrôle excite chaque broche d'entrée à la valeur VIH ou VIL définie pour cette broche sur la base de cette donnée.

VOH et VOL sont les niveaux de comparaison des broches de sortie. Un niveau "1" correspond à une valeur supérieure à VOH. Un niveau "0" correspond à un niveau inférieur à VOL. Les niveaux de sortie sont déterminés par des comparateurs disposés dans l'équipement et couplés respectivement aux broches de sortie définies.

MODIFICATION DE LA SEQUENCE (DEROULEMENT) DES SEGMENTS DE TESTS.

En plus de pouvoir examiner et modifier les paramètres d'un segment de test existant, on peut aussi modifier la succession logique de segments de test. Un exemple est donné par l'introduction d'un nouveau segment de test dans une séquence existante d'un programme de contrôle. La figure 9 est une image-écran "FlowTool" (outil de déroulement), qui illustre graphiquement une séquence de tests existante et un nouveau segment de test 910, qui n'a pas encore été logiquement relié à la séquence de tests existante. Le nouveau segment de test 910 est créé et connecté dans la séquence de test comme décrit ci-dessous.

Le fait d'activer le bouton droit de la souris alors que le curseur est dans l'aire de fond 920 constituant la partie inutilisée de l'image-écran a pour effet d'amener l'affichage d'un menu principal, tel que représenté sur la figure 10. Le premier article du menu principal est "segments"(segments). Le fait de sélectionner l'option segments produit un menu de segments 1010. Le menu de segments 1010 affiche un groupe de choix pour les différents types de segments de tests. Le premier choix, à la ligne 1020, est "FunctionalTest". En sélectionnant "FunctionalTest", on provoque l'affichage d'une nouvelle icône "FunctionalTest" 910 dans l'image-écran de l'outil de déroulement représentée sur la figure 9. A ce moment, l'icône ne présente aucune interconnexion avec le programme de contrôle, ni de données associées définissant le segment de

test. On peut introduire le critère de réussite ou d'échec relatif à ce nouveau segment de test en sélectionnant l'icône de test 910, ce qui conduit à l'ensemble image-écran et interface de l'outil "FunctionalTest" qui a été décrit ci-dessus en relation avec la figure 6. L'outil "FunctionalTest" produit l'interface servant à définir le test de fonctionnement associé à l'icône nouvellement créée 910.

On se reporte à la figure 9. L'intégration de l'icône "FunctionalTest" 910 dans la séquence de tests s'effectue de la manière suivante. Le port 330 de l'icône de test de continuité 320 est, à ce moment, connecté à l'icône d'arrêt 340. Le fait d'enfoncer le bouton gauche sur le port 310 coupe cette connexion, en amenant l'extrémité gauche de la ligne de connexion 345 à se déplacer avec le curseur. Ceci permet effectivement à l'utilisateur de "capturer et garder" l'extrémité de la ligne de connexion 345 qui était antérieurement connectée au port 330. Le fait de lâcher le bouton gauche amène la ligne de connexion 345 à se "rétracter" dans l'icône d'arrêt 340 d'une manière qui rappelle visuellement la rétractation du cordon d'alimentation électrique d'un aspirateur.

L'enfoncement du bouton sur le port 330 de l'icône de test de continuité 320 fait qu'une nouvelle ligne de connexion est extraite du port 330. Une extrémité de cette nouvelle ligne de connexion est attachée au port 330 et l'autre extrémité est attachée au curseur. On peut déplacer l'extrémité de la ligne de connexion attachée au curseur de manière à l'amener à proximité immédiate d'un autre port, par exemple le port 330 de l'icône de test 910, après quoi on la libère. Lorsqu'on libère l'extrémité de la ligne de connexion à proximité immédiate d'un port, celle-ci se connecte automatiquement, ce qui définit un nouveau déroulement de séquence de tests. Plus spécialement, si les paramètres de réussite ou d'échec du test de continuité 320 ne sont pas satisfaits, alors, au lieu que la séquence de tests s'arrête, le test de fonctionnement correspondant à l'icône de test 910 est exécuté.

La nouvelle ligne de connexion 1110 est illustrée sur la figure 11.

De la même façon, une autre ligne de connexion 1120 peut être tirée, à l'aide du curseur, du port 1130 de l'icône de test 910 jusqu'au port 1140 de l'icône "power" (alimentation électrique) 1150, par libération de l'extrémité, commandée par le curseur, de la ligne de connexion 1120 au voisinage immédiat du port 1140 de l'icône "Power" 1150. Enfin, la connexion du port 1160 du segment de test 910 peut être établie avec le port de l'icône "Stop" (arrêt) 340.

La figure 12 montre la nouvelle icône de test 910 entièrement connectée, laquelle comporte une nouvelle ligne de connexion 1210 entre le port 1160 de l'icône de test 910 et le port 1220 de l'icône d'arrêt 340. Dans cette nouvelle séquence, si le test de continuité qui est représenté par l'icône 320 est satisfait, la séquence de tests passe au segment de test de fuite représenté par l'icône "Leakage" 325. Dans le cas où le test de continuité n'est pas satisfait, la séquence de tests passe au segment de test représenté par l'icône 910. Dans le cas où le segment de test représenté par l'icône de test 910 est satisfait, la séquence de tests passe à un segment de test d'alimentation électrique représenté par l'icône "Power" 1150 et "saute" le test de fuite représenté par l'icône "Leakage" 325. Dans le cas où le segment de test représenté par l'icône de test 910 n'est pas satisfait, la séquence de test passe à l'icône "Stop" 340 et s'arrête. De cette manière, on peut créer un nouveau segment de test et l'introduire dans une séquence de programmes de tests existante. Il est également possible de modifier le déroulement d'une séquence de tests existante, ou bien de créer une nouvelle séquence de tests.

AUTRES OUTILS ET CONVENTIONS RELATIVES A LA SOURIS.

Dans les différents outils, les broches se trouvant sur le dispositif soumis au test sont désignées par des noms logiques, et non par des numéros de broches. Par exemple, des noms logiques, appelés "Pindefs", sont utilisés par l'interface "DCTool" qui est illustrée dans la fenêtre 520 de la figure 5. Un programme "Pindef" (définition de broche) fournit une interface graphique (non illus-

trée) permettant de définir des noms de broches en relation particulière avec les numéros des broches.

La souris à trois boutons qui est utilisée pour commander le curseur est le dispositif permettant de pointer, de capturer, de traîner et de sélectionner. Le bouton droit de la souris sert à faire apparaître des menus en fonction du contexte. Ainsi, un menu qui est propre à l'objet pointé par le curseur apparaîtra lorsqu'on enfoncera le bouton droit. De nombreux outils possèdent un "outil de menu principal" qui apparaît lorsqu'on enfonce le bouton droit sur une aire quelconque inutilisée du fond. Le bouton du milieu sert à tirer, à déplacer ou à effectuer un panoramique. On accomplit ceci en plaçant le curseur sur l'objet à déplacer, en pressant le bouton du milieu, en déplaçant le curseur jusqu'à la position voulue et en relâchant le bouton du milieu. Le bouton gauche sert à effectuer une capture, une sélection, un choix ou à activer.

LE SYSTEME DE CONTROLE.

Les outils d'interface du poste de travail 100 ainsi que les programmes et les données de l'ordinateur du système de contrôle 130 constituent un programme de contrôle du type "résultant" non spécialisé. Chaque test de fonctionnement est représenté graphiquement sur le poste de travail 100 par une icône. L'ordre de succession de ces tests de fonctionnement est indiqué par les interconnexions graphiques des icônes. Chaque test de fonctionnement est également représenté par des codes d'exécution de données correspondants du système de contrôle, et la succession de ces tests de fonctionnement est représentée par des pointeurs reliant logiquement les codes d'exécution de données correspondants, comme décrit plus complètement ci-dessous.

L'architecture de l'ordinateur du système de contrôle 130 est illustrée sur la figure 13. Le poste de travail 100 est couplé à un clavier 110 et à une souris 120 et comporte des outils d'interface 1305 qui commandent le dispositif d'affichage 1310 et répondent à des manoeuvres appliquées au clavier 110 et à la souris 120. Des messages sont envoyés des outils d'interface 1305 à des programmes d'interface d'outil 1315 inclus dans l'ordinateur 130 du

système de contrôle afin de modifier les codes d'exécution de données du système de contrôle pour qu'ils correspondent aux paramètres des tests de fonctionnement et aux séquences de tests définis par les interfaces apparaissant sur l'écran 1310. Des messages sont également envoyés pour faire commencer des actions particulières, par exemple à un message demandant d'"exécuter" un test, et des données et des résultats de test peuvent être envoyés au poste de travail 100 par l'ordinateur 130 du système de contrôle pour y être affichés.

10 Dans le mode de réalisation préféré, les outils d'interface 1305 comprennent un ensemble matériel et logiciel normalisé de traitement graphique à X fenêtres et des outils d'interface spécifiques, par exemple le programme "FlowTool" et les autres programmes décrits ci-dessus. Les outils d'interface 1305 définissent des
15 icônes et des séquences d'icônes sous la forme de noms d'icônes pouvant être définis par l'utilisateur, comme par exemple "Continuity". Ainsi, lorsqu'un message est envoyé aux programmes d'interface d'outil 1315 de l'ordinateur 130 du système de contrôle en réponse à la création, à la modification ou à la reconnexion d'une
20 icône, le message est donné en termes de nom d'icône.

Egalement selon le mode de réalisation préféré de l'invention, les programmes d'interface d'outil 1315 interprètent les messages se rapportant aux icônes au moyen d'une table binaire 1320 servant à traduire les noms d'icônes en adresses des blocs de segments correspondants contenus dans la mémoire de l'ordinateur 130 du système de contrôle. Les programmes d'interface d'outil 1315 examinent alors ou modifient les codes d'exécution de données correspondants en réponse aux messages. Ainsi, lorsque l'utilisateur déplace une ligne interconnectant des icônes à l'aide du programme "FlowTool", les noms des icônes considérées sont recherchés
30 dans la table binaire 1320 afin que soient déterminées les adresses des blocs de segments correspondants. Les programmes d'interface d'outil 1315 modifient alors les pointeurs appropriés de façon qu'ils correspondent à la nouvelle séquence définie par la nouvelle position de la ligne d'interconnexion. Ces codes d'exécution de
35 données et leurs relations avec les icônes de l'interface et

d'autres codes d'exécution de données sont discutés plus complètement ci-dessous.

Les programmes d'interface d'outil 1315 interprètent et mettent en oeuvre également des instructions telles que "exécution". Par exemple, les programmes d'interface d'outil 1315 appellent un programme d'exécution placé dans les programmes d'exécution de contrôle 1325 en réponse à une instruction d'"exécution" venant des outils d'interface 1305, par exemple par sélection du bouton "Execute" (exécution) de l'aire 505 de l'interface "FunctionalTest" représentée sur la figure 6. De plus, les programmes d'interface d'outil 1315 transmettent des messages de données au poste de travail 100 en vue de leur affichage sous forme graphique. Par exemple, dans le mode de réalisation préféré, les lignes d'interconnexion deviennent blanches lorsqu'il s'agit de représenter la séquence de tests réelle pendant l'exécution des tests. En outre, les bords des icônes correspondant aux tests exécutés changent de couleur pour indiquer si les tests ont été satisfaits ou non relativement aux critères imposés pour les tests respectifs.

Chaque test de fonctionnement est défini principalement par trois types distincts de codes d'exécution de données. Tout d'abord, un ou plusieurs blocs de données de paramètres de test, par exemple le bloc de données 1330, définissent des paramètres de fonctionnement pour chaque test de fonctionnement. Les blocs de données ne contiennent aucune information de code ou de séquence exécutables, mais sont spécialement destinés aux données définissant les paramètres d'un test spécifique. Typiquement, ces paramètres de fonctionnement sont définis, examinés et modifiés en fonction d'éléments introduits dans les outils correspondants. Par exemple, des modifications d'un test de courant continu seraient obtenues par application de messages au programme "DCTestTool" (décrit ci-dessus en relation avec la figure 5), des modifications à un test de fonctionnement seraient obtenues par application de messages à l'outil "FunctionalTest" (décrit ci-dessus en relation avec la figure 6), et des modifications à la synchronisation d'un test défini seraient obtenues par application de messages au pro-

gramme "TimingTool" (décrit ci-dessus en relation avec la figure 7).

05 Les blocs de données peuvent être modifiés de deux manières. Tout d'abord, on peut sélectionner un bloc différent. Par exemple, le bloc "grossfuncTIM006" de la ligne "Timing" (figure 7) pourra être remplacé par un bloc différent par sélection d'un bloc de données dans l'ensemble image-écran et interface "SelectTool" illustré sur la figure 1. Plus spécialement, en enfonceant le bouton droit alors que le curseur est sur une appellation de zone, par
10 exemple l'appellation "Timing" 618 de la ligne 620, on produit un menu permettant de sélectionner un bloc de données différent pour définir la synchronisation. Sinon, on peut modifier des paramètres spécifiques se trouvant à l'intérieur d'un bloc de données sélectionné en utilisant l'outil de l'ensemble image-écran et interface
15 approprié. Plus spécialement, en activant le bouton gauche alors que le curseur est sur l'aire du bloc de données, on "ouvre" l'outil correspondant à l'attribut sélectionné.

Un deuxième type de code d'exécution de données est un bloc de segment. Un exemple est illustré sur la figure 13 sous la
20 forme du bloc de segment 1335. Le troisième type de code d'exécution de données est une définition de segment, dont un exemple est représenté par la définition de segment 1340 de la figure 13. Les blocs de segment et les définitions de segment définissent ensemble les séquences logiques des tests de fonctionnement qui sont disponibles au moment de l'exécution.
25

Les blocs de segment et les définitions de segment pourront être expliqués de façon plus complète en relation avec la figure 14. Comme illustré, le bloc de segment 1335 est un code d'exécution de données possédant un certain nombre de positions.
30 Par exemple, la position "nom" contient le nom de l'icône correspondante de l'écran-image 1310. La position "type" identifie le type de l'objet, à savoir "segment", qui constitue le bloc de segment. La position "autre type" identifie quel type de test est pointé, à savoir "le test de courant continu". D'autres positions
35 indiquent comment d'autres séquences sont rendues disponibles (lesquelles correspondent au nombre de ports de sortie présents sur

l'affichage graphique) et l'adresse particulière de la définition de segment 1340 qui contient les adresses d'autres blocs de segment 1410 et 1420 qui définissent les autres tests de fonctionnement disponibles après l'exécution du test de courant continu défini par le bloc de segment 1325 au moment de l'exécution. Dans le présent exemple, les blocs de segment 1325, 1410 et 1420 correspondent aux icônes 320, 425 et 340 des figures 3 et 4. (Les icônes composites sont principalement un mécanisme d'interface des outils d'interface 1305 du poste de travail 100, bien qu'ils soient représentés dans l'ordinateur 130 du système de contrôle par des codes d'exécution de données correspondants).

On revient à la figure 13. L'ordinateur 130 du système de contrôle contient de préférence des programmes d'exécution de test 1325 écrits dans le langage de programmation C. Les programmes d'exécution de test 1325 constituent une bibliothèque de support d'exécution de procédés prédéfinis qui exécutent les blocs de segment, les définitions de segment et les blocs de données et qui constituent une interface vis-à-vis de l'équipement de contrôle 135 pour l'exécution du déroulement du contrôle correspondant à la représentation graphique présente sur l'écran-image en réponse à des messages reçus de la part des programmes d'interface d'outil 1315. Par exemple, en réponse à un message demandant l'exécution d'un test, des programmes d'interface d'outil 1315 identifient le test à exécuter à l'aide d'une adresse spécifique et appellent les programmes d'exécution de test 1325. Les programmes d'exécution de test constituent alors l'interface avec les blocs de segment appropriés, les définitions de segment, les blocs de données et l'équipement de contrôle 135. Par exemple, en réponse à un message demandant d'exécuter un test de courant continu, les programmes d'exécution de test lus dans le bloc de segment adressé exécutent un programme d'exécution approprié à l'exécution d'un test de courant continu, lisent les paramètres du bloc de données définissant le test de courant continu, excitent l'équipement de contrôle 135 en fonction de ces paramètres de test, comparent les résultats de test reçus de la part de l'équipement de contrôle 135 avec les paramètres de réussite ou d'échec contenus dans le bloc de données,

et poursuivent la séquence logique jusqu'au bloc de segment suivant de la séquence de tests en réponse au contenu logique de la définition de segment correspondante.

05 L'équipement de contrôle 135 comporte des dispositifs d'excitation qui sont couplés au dispositif soumis au test et servent à appliquer des signaux d'entrée définis au circuit intégré soumis au test, ainsi que des comparateurs servant à comparer les signaux de sortie du dispositif soumis au test avec des niveaux programmables. Lorsqu'un test est en cours d'exécution, les données
10 reçues de la part de l'équipement de contrôle sont comparées avec les paramètres de réussite et d'échec, et les résultats sont emmagasinés dans une aire de données brutes. Il peut être fait accès à ces données à l'aide des programmes d'interface d'outil 1315, et celles-ci sont envoyées sous forme de message au poste de travail
15 100 pour être affichées sur l'écran 1310.

Bien entendu, l'homme de l'art sera en mesure d'imaginer, à partir du système dont la description vient d'être donnée à titre simplement illustratif et nullement limitatif, diverses variantes et modifications ne sortant pas du cadre de l'invention.

REVENDEICATIONS

1. Système de contrôle permettant de soumettre un dispositif à une séquence logique de tests de fonctionnement, où la séquence est définie par une interface graphique, le système comportant un moyen d'affichage, un moyen servant à appliquer des signaux au dispositif ; un moyen servant à recevoir des signaux de la part du dispositif ; un moyen servant à traiter les signaux reçus du dispositif, et un moyen servant à définir les paramètres des tests de fonctionnement, qui comportent des signaux à appliquer au dispositif et les signaux devant être reçus de la part du dispositif, le système étant caractérisé en ce qu'il comprend en outre :

un moyen (305) servant à représenter graphiquement sur le moyen d'affichage (1310) une séquence de tests de fonctionnement ;

une structure de données (1330) définissant une séquence de tests de fonctionnement ;

un moyen (1305) servant à manipuler la structure de données (1330) pour établir la correspondance avec la séquence affichée ; et

un moyen (130) servant à commander le système afin que soient exécutés les tests de fonctionnement contenus dans la séquence définie par la structure de données.

2. Système de contrôle selon la revendication 1, caractérisé en ce qu'il comprend en outre un moyen (1315) servant à modifier la séquence graphiquement représentée sur le moyen d'affichage (1310), en ce que ledit moyen de manipulation modifie la structure de données (1330) pour établir la correspondance avec la séquence affichée, et en ce que ledit moyen de commande fait que l'équipement de contrôle exécute la séquence modifiée.

3. Système de contrôle selon la revendication 1, caractérisé en ce que la structure de données comprend un certain nombre de codes d'exécution de données (1330) indépendants qui peuvent être arbitrairement interconnectés afin de définir une séquence de tests de fonctionnement.

4. Système de contrôle selon la revendication 3, caracté-

térisé en ce que l'information de la séquence qui est relative à chaque test de fonctionnement est représentée par un code d'exécution de données distinct.

5 5. Système de contrôle permettant de soumettre un dispositif à une séquence logique de tests de fonctionnement, où la séquence est définie par une interface graphique, le système comportant un moyen servant à appliquer des signaux au dispositif, un moyen servant à recevoir des signaux de la part du dispositif, un moyen servant à traiter les signaux reçus de la part du dispositif,
10 un moyen servant à définir des paramètres des tests de fonctionnement, qui comprennent les signaux à appliquer au dispositif et les signaux devant être reçus du dispositif, et un moyen servant à examiner et à modifier les paramètres, le système comprenant en outre une interface graphique servant à examiner et modifier la
15 séquence logique des tests de fonctionnement, caractérisé en ce que l'interface comprend :

 un moyen d'affichage graphique (1310) possédant un curseur (205) et un dispositif d'entrée (110, 120) servant à commander la position du curseur sur le moyen d'affichage
20 graphique ;

 un moyen (305) servant à produire une représentation graphique de chaque test de fonctionnement sur le moyen d'affichage graphique (1310) ;

 un code d'exécution de données distinct correspondant à
25 chaque test de fonctionnement, le code d'exécution de données comportant des pointeurs destinés à se pointer sur des codes d'exécution de données correspondant à d'autres tests de fonctionnement séquentiels ;

 un moyen servant à représenter graphiquement l'adresse de
30 chaque test de fonctionnement en association avec l'icône graphique correspondante ;

 un moyen servant à interconnecter graphiquement, à l'aide de lignes (1110, 1120, 1210), des icônes (310, 320, 325, 910) se trouvant sur le moyen d'affichage, les lignes correspondant aux
35 cheminements possibles de la séquence de test ;

 un moyen de données qui répond au moyen servant à connec-

5 ter graphiquement des icônes (310, 320, 325, 910) présentes sur le moyen d'affichage (1310) en définissant les pointeurs correspondants dans les codes d'exécution de données de façon que les autres tests de fonctionnement séquentiels possibles définis par les codes d'exécution de données correspondent aux cheminement possibles de la séquence de tests graphiquement affichée ; et

10 un moyen de contrôle (135) couplé au dispositif (140) et servant à exécuter la séquence logique de tests de fonctionnement à partir des séquences de tests possibles, en réponse à des signaux reçus de la part du dispositif (140) et aux codes d'exécution de données.

15 6. Système selon la revendication 5, caractérisé en ce que ledit moyen (120) servant à interconnecter graphiquement des icônes comprend en outre des moyens servant à modifier les interconnexions entre icônes et en ce que ledit moyen de données modifie les codes d'exécution de données de façon qu'ils correspondent à la représentation graphique.

20 7. Système selon la revendication 6, caractérisé en ce qu'il comprend en outre un moyen d'entrée (120) servant à déconnecter les lignes entre icônes et à tracer de nouvelles lignes entre icônes, ces deux opérations s'effectuant sous commande d'un curseur.

25 8. Système selon la revendication 7, caractérisé en ce qu'il comprend en outre un certain nombre de types de tests de fonctionnement prédéfinis, chaque type de test de fonctionnement étant représenté graphiquement sur le moyen d'affichage (1310) par une icône commune (325), chaque type de test de fonctionnement étant également représenté par une classe correspondante de codes d'exécution de données, et chaque test de fonctionnement spécifique
30 étant représenté par un élément de cette classe.

35 9. Système selon la revendication 5, caractérisé en ce que ledit moyen (305) servant à produire une représentation graphique comprend en outre un moyen servant à représenter des groupes de tests de fonctionnement sous la forme d'une icône unique (325).

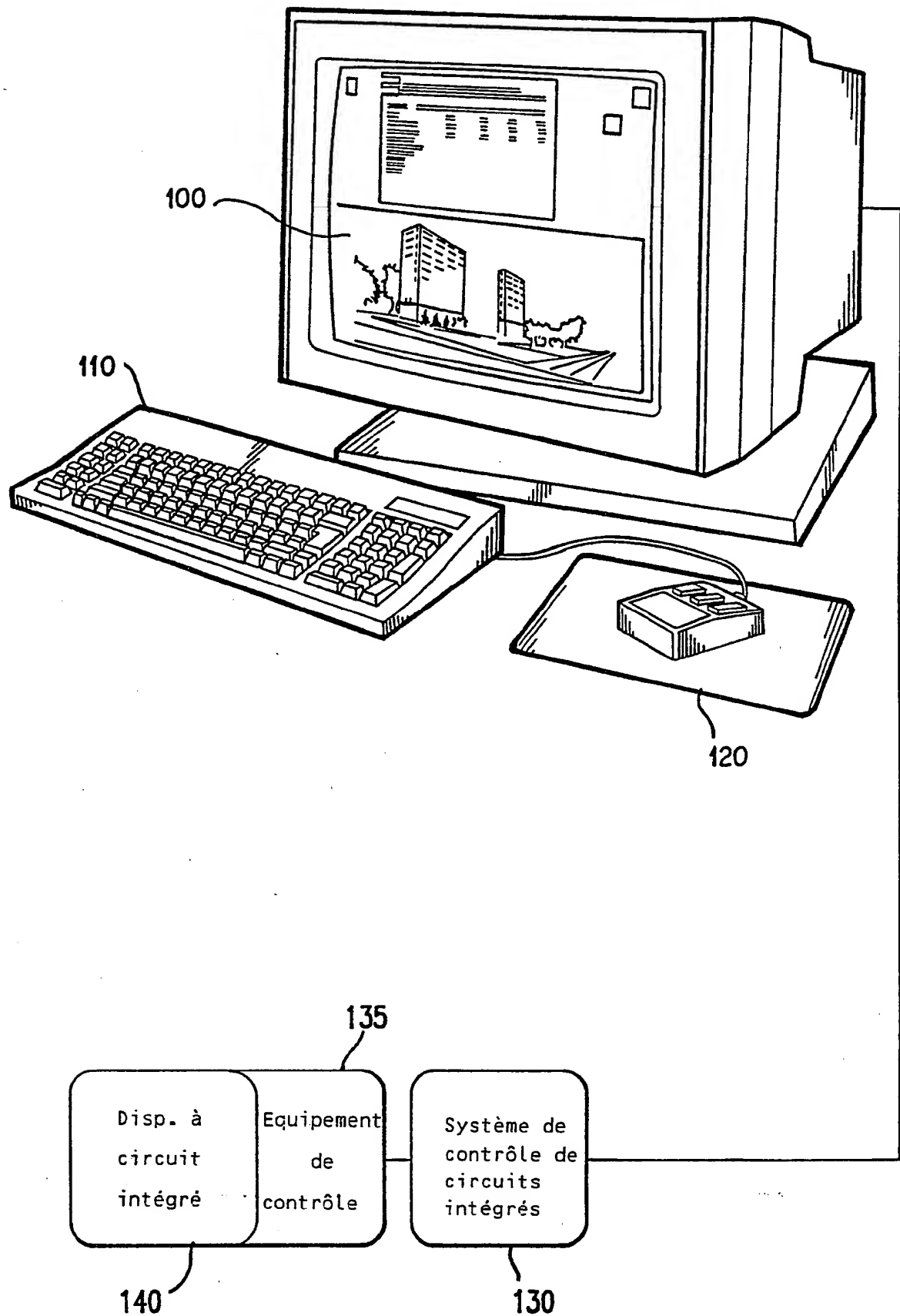


FIG. 1

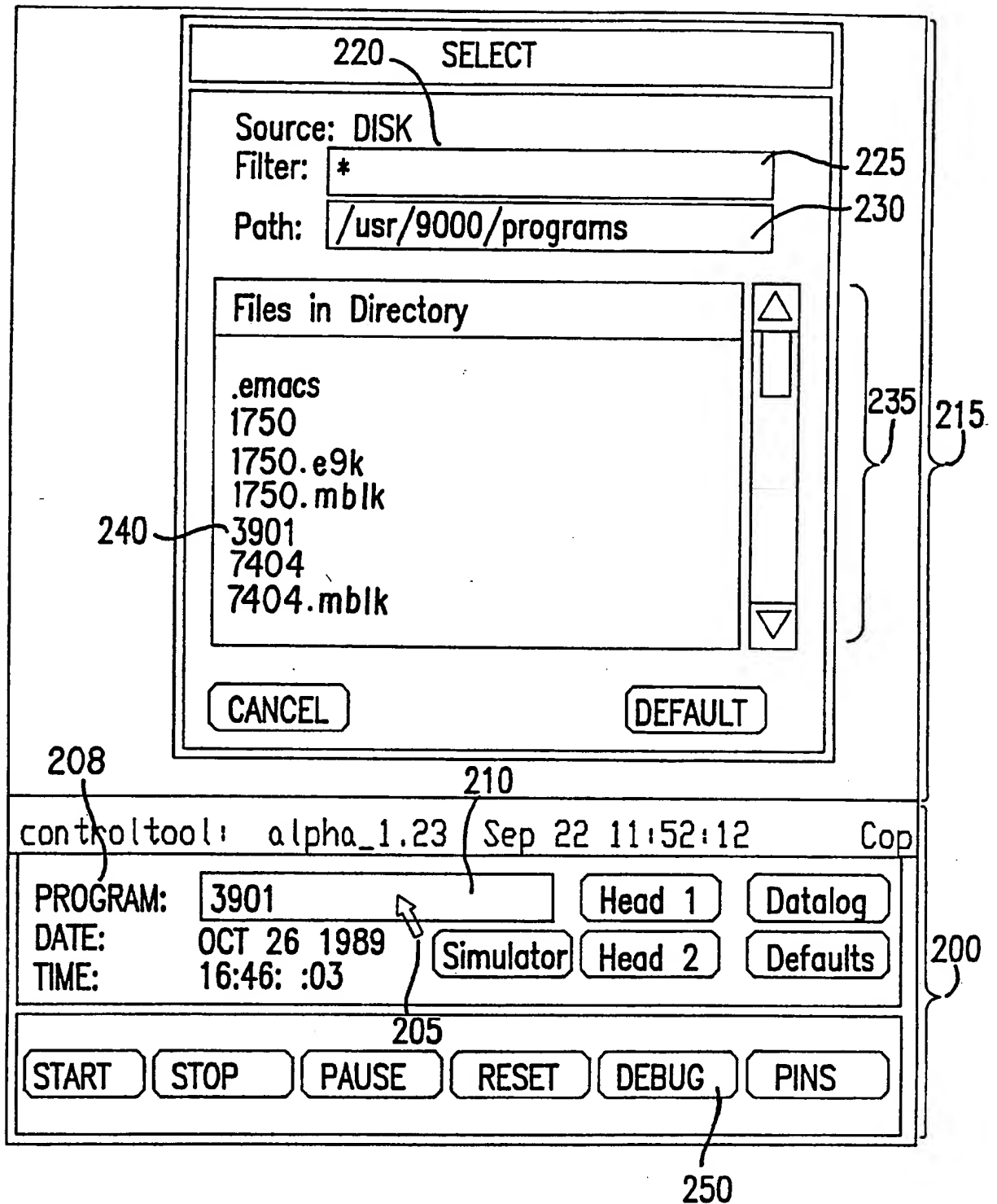


FIG. 2

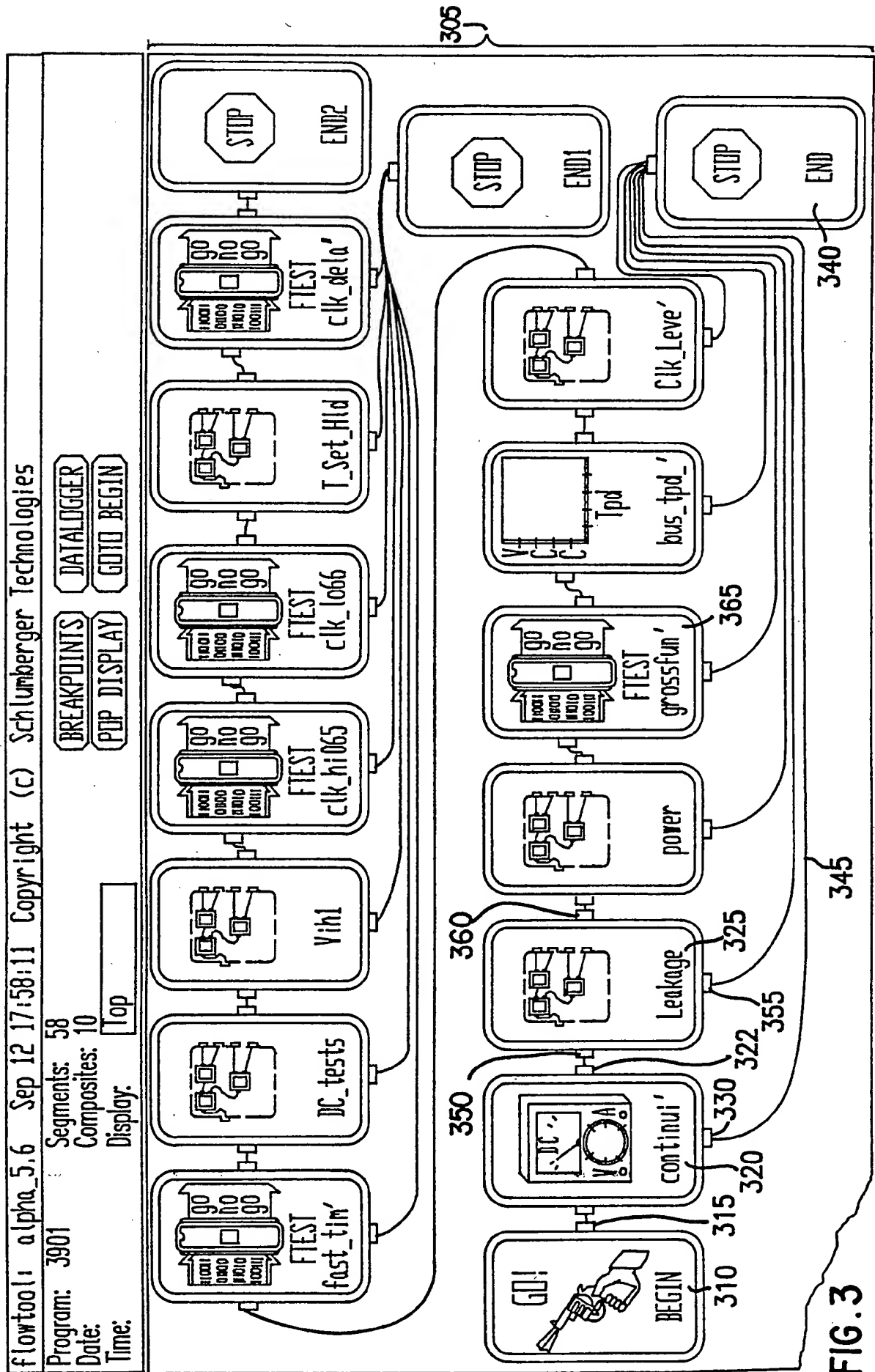


FIG. 3

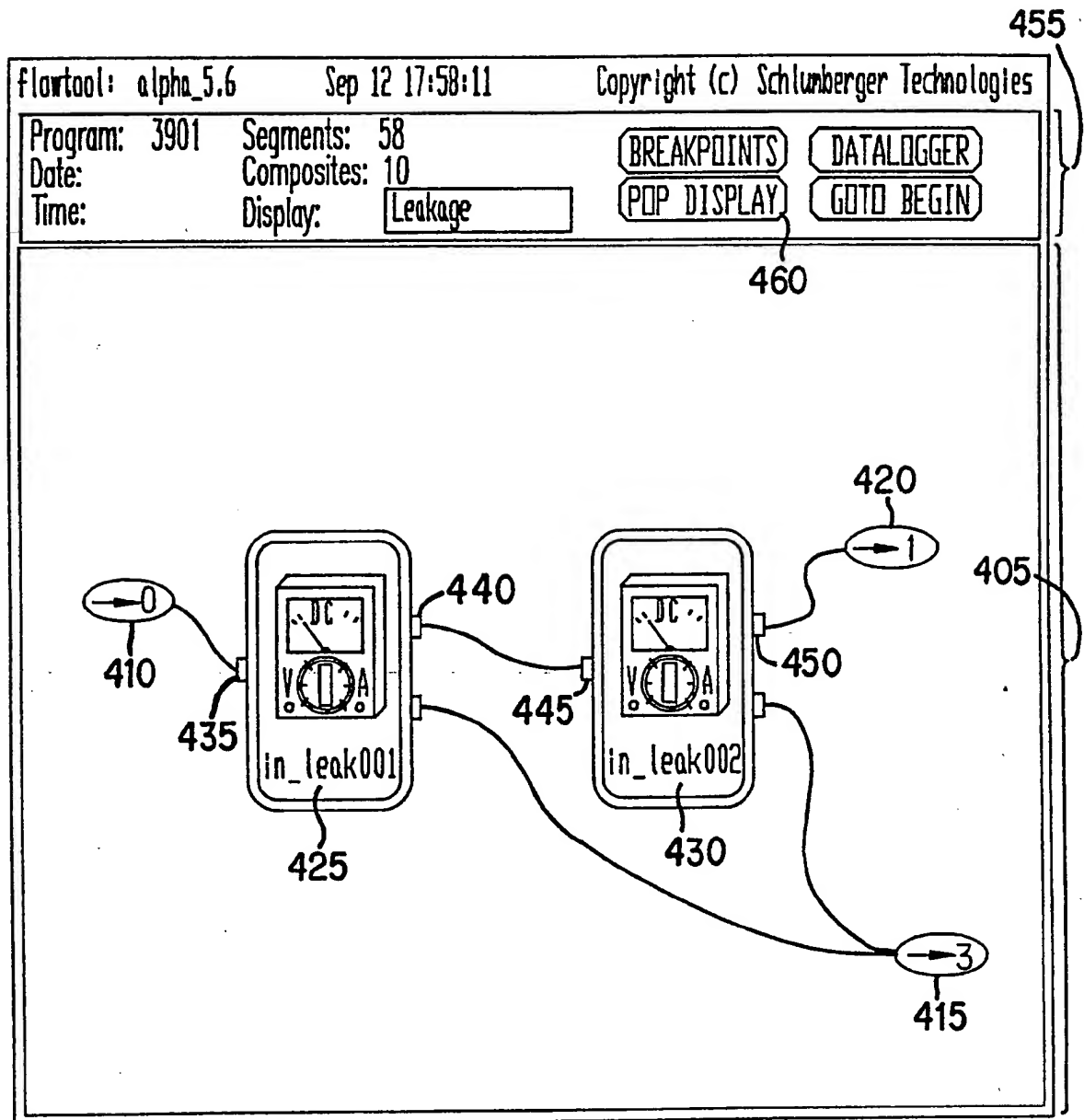


FIG. 4

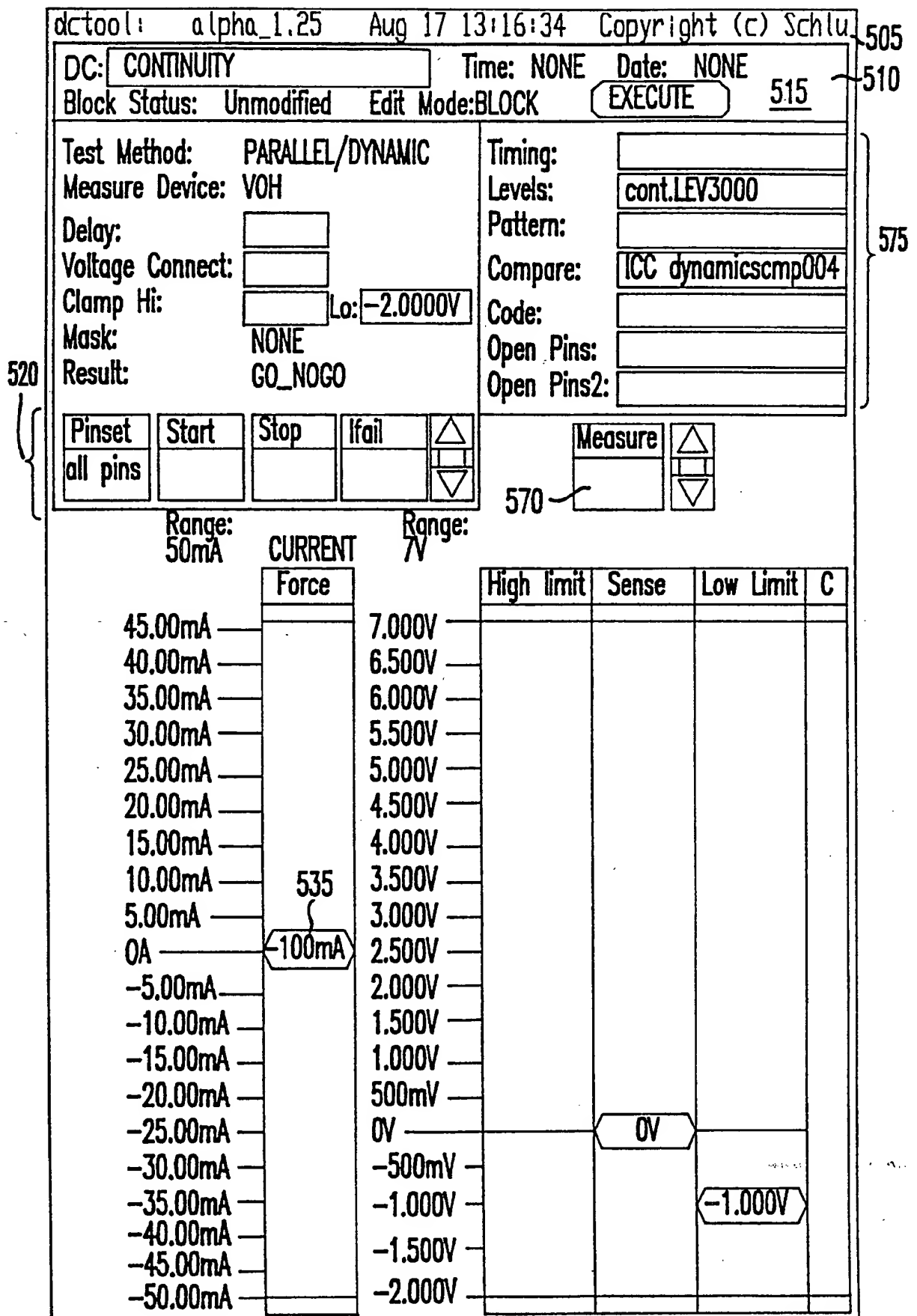


FIG. 5

530

540

550

560

| ftesttool: alpha_1.12 Sep 13 | |
|------------------------------|----------------------------------|
| Ftest: grossfunc 006 | |
| Date: NONE | 610 Edit Mode: BLOCK |
| Time: NONE | |
| Status: UNMODIFIED | EXECUTE |
| 605 | |
| Timing: grossfuncTIM006 | 620 |
| Levels: ICC_quiesLEV003 | 630 |
| Code: | |
| Open Pins: | |
| 615 | |
| Vector: BIT001 | 635 |
| Pattern ΔEQ | Start |
| Stop | Compare 02XHA |
| | <input type="button" value="△"/> |
| | <input type="button" value="▽"/> |
| 625 | |

FIG. 6

| | | | | | |
|-----------------------------|---------------|----------------------|--|-------------------------|--|
| timingtool: Alpha2.0 | | Sep 26 10:44:24 | | Copyright (c) Schlumber | |
| TIMING: | | grossfuncTIM006 | | DATE: NONE | |
| PINDEF TABLE: | | | | TIME: NONE | |
| BLOCK STATUS: UNMODIFIED | | TIMING TYPE: | | EDIT MODE: BLOCK | |
| DISPLAY MODE: PERIOD EVENTS | | 735 | | 740 | |
| DEPENDENT EDGE: FLOATING | | VARIABLES | | CONSTANTS | |
| | | TEST | | CHECK | |
| SCALE | PERIOD NAME | TO | | | |
| 1 TIMES | p0 | | | | |
| PINDEF | SEQUENCE NAME | WAVEFORM DESCRIPTION | | | |
| a | t0f0d | 725 | | | |
| b | t0f0d | | | | |
| cp | t0f0d | | | | |
| ih | t0f0d | | | | |
| im | t0f0d | | | | |
| il | t0f0d | | | | |
| d | t0f0d | | | | |
| y | t0f0t | 765 | | | |
| 745 | 750 | 760 | | | |
| | | 755 | | | |

FIG. 7

leveltool: Proto_1.7 Sep 13 16:39:56 Copyright (c) Schlumberger

Level Block:

Edit Mode: HARDWARE Block Status: UNMODIFIED

Display Mode: LEVEL Date: NONE Time: NONE

| VIH | VIL | VOH | VOL | Vref | -IOH | IOI |
|-----|-----|-----|-----|------|---------|-----|
| 840 | 850 | 860 | 870 | 880 | 50.00mA | |
| | | | | | 45.00mA | |
| | | | | | 40.00mA | |
| | | | | | 35.00mA | |
| | | | | | 30.00mA | |
| | | | | | 25.00mA | |
| | | | | | 20.00mA | |
| | | | | | 15.00mA | |
| | | | | | 10.00mA | |
| | | | | | 5.00mA | |
| | | | | | 0A | |

| Pinsets | VIH | VIL | VOH | VOL | Vref | -IOH | IOI |
|------------|--------|-------|--------|--------|------|------|-----|
| Vdd<> | 2.379V | 400mV | - | - | - | - | - |
| co<> | 2.949V | 0V | - | - | - | - | - |
| all_inputs | 2.799V | 0V | 1.501V | 1.501V | - | - | - |
| pins_to | | | | | | | |
| all_output | | | | | | | |

FIG. 8

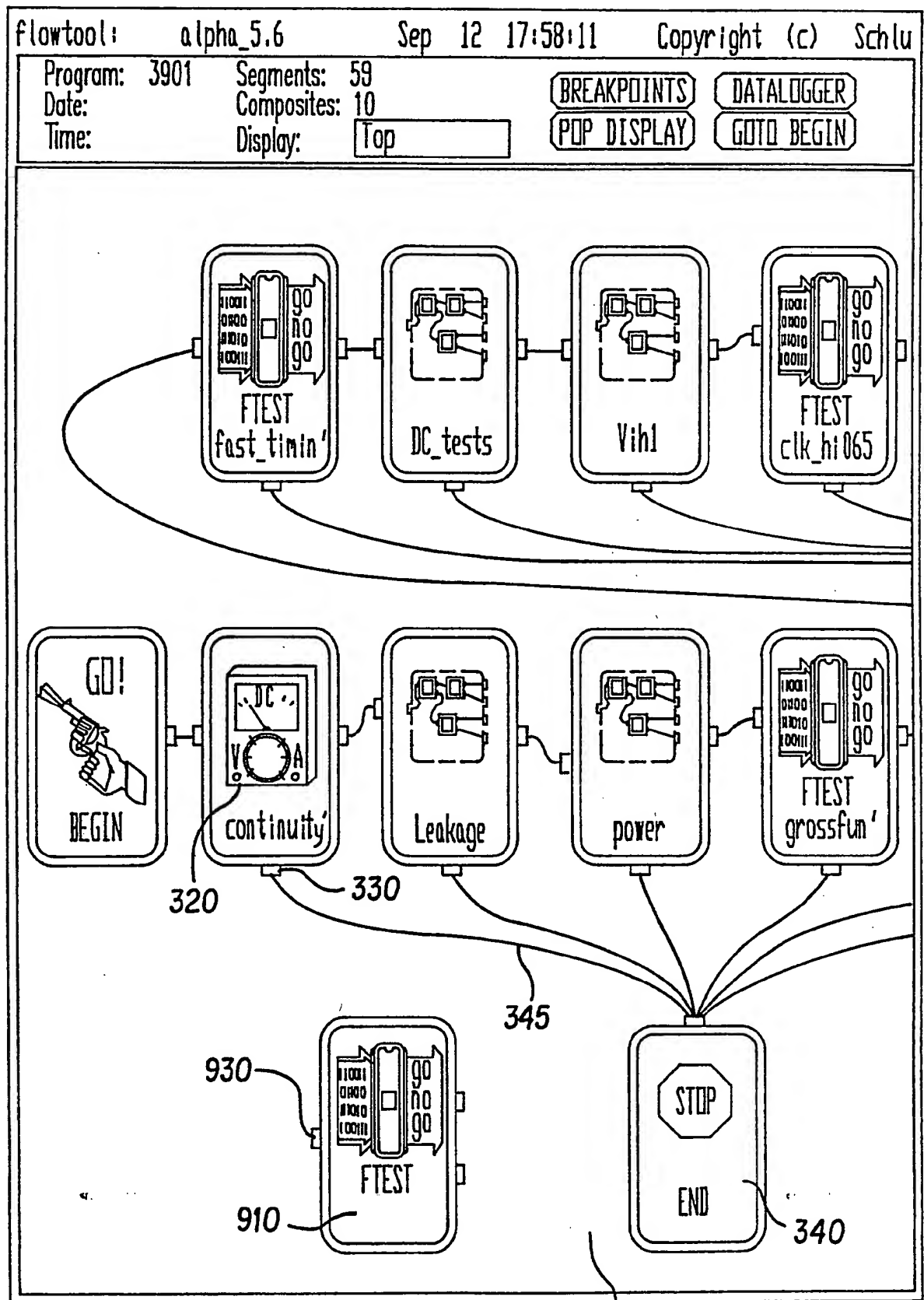


FIG. 9

920

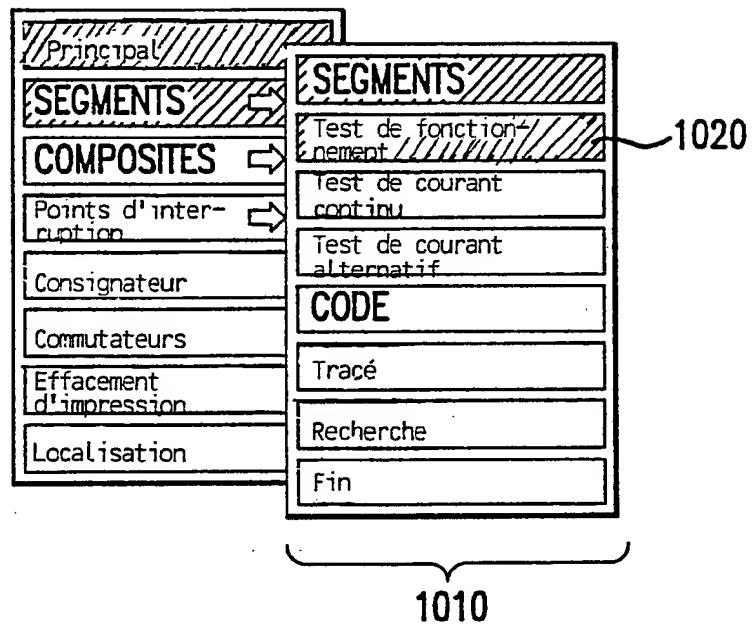


FIG. 10

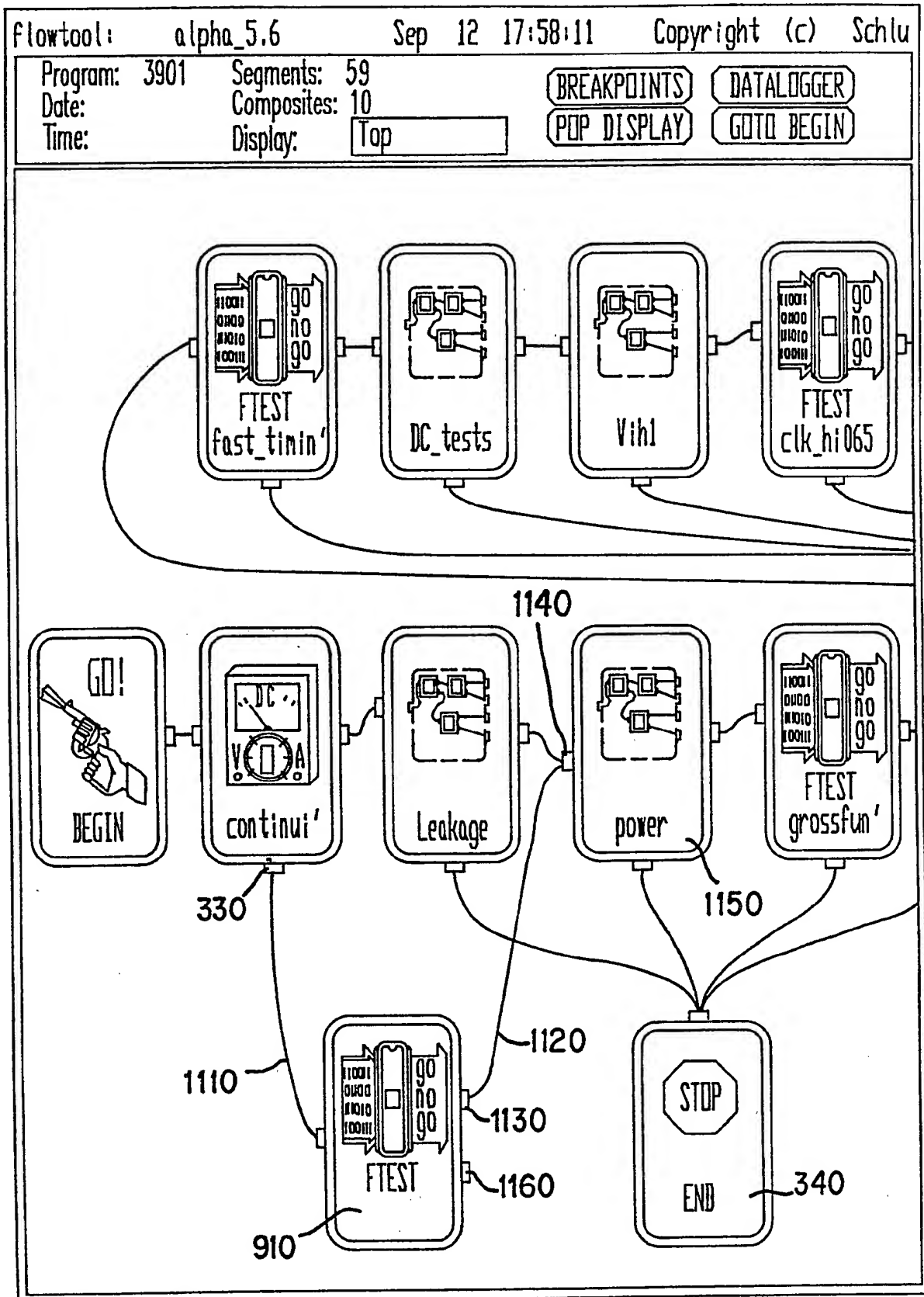


FIG. 11

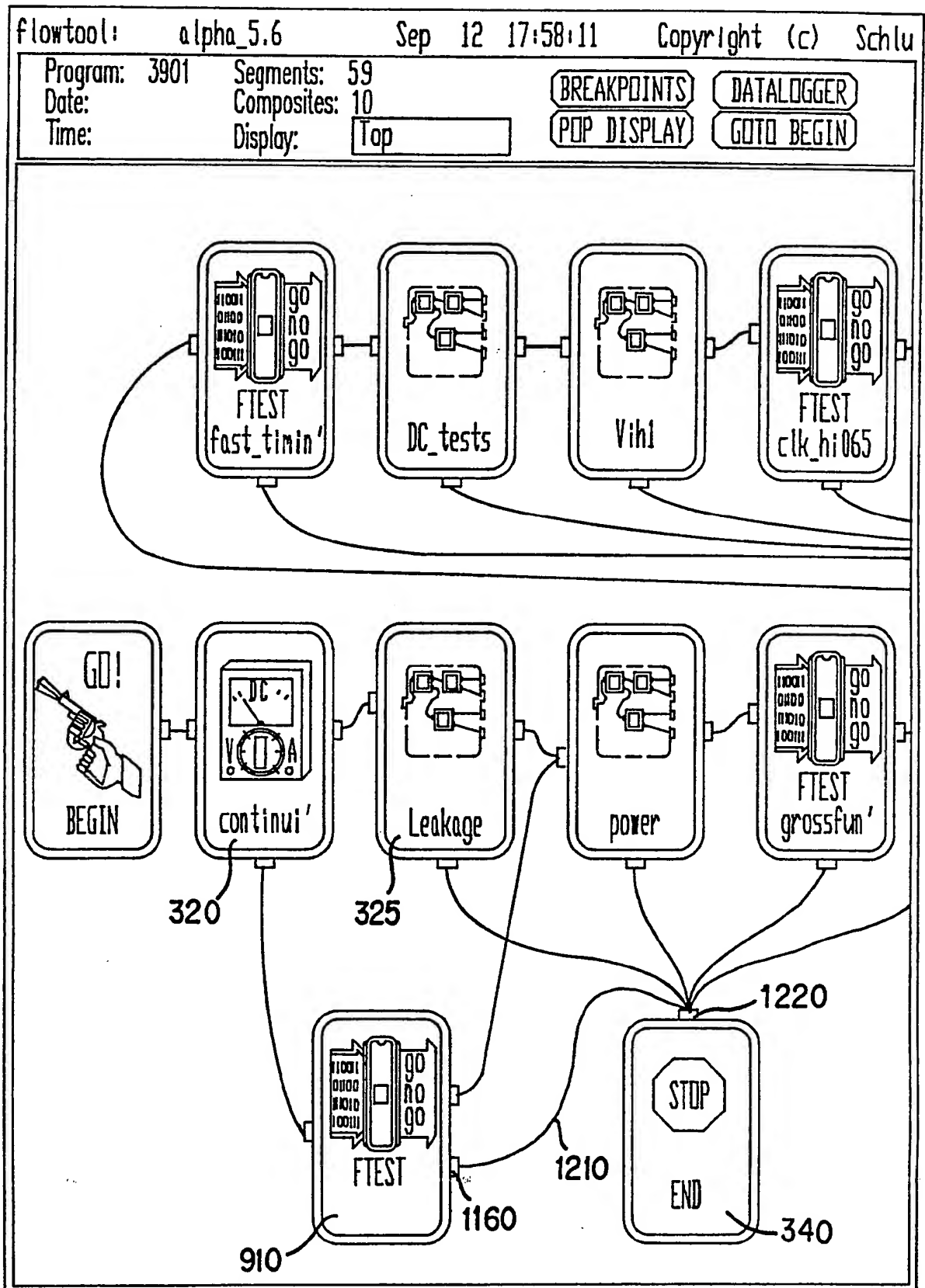


FIG. 12

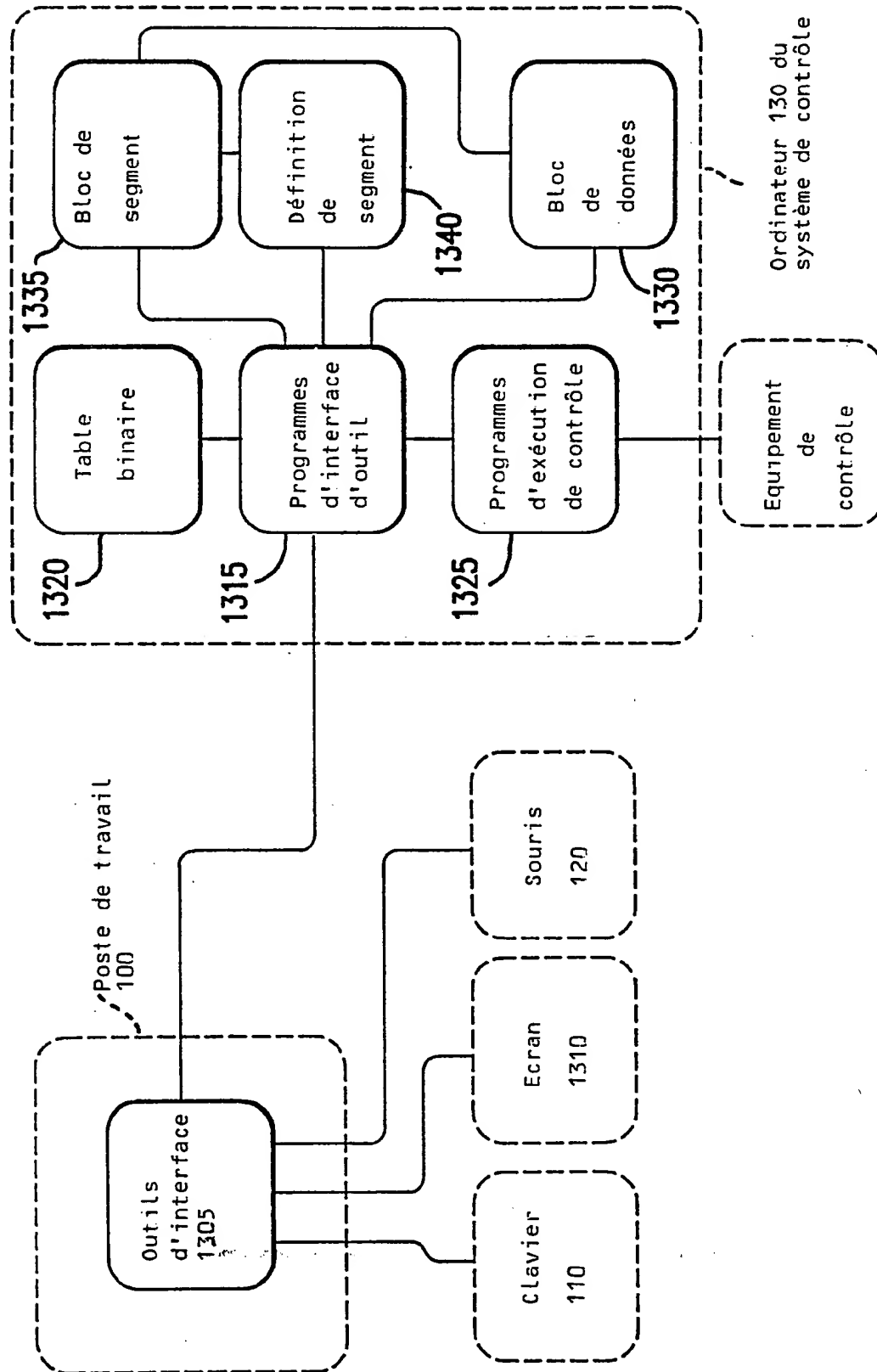


FIG. 13

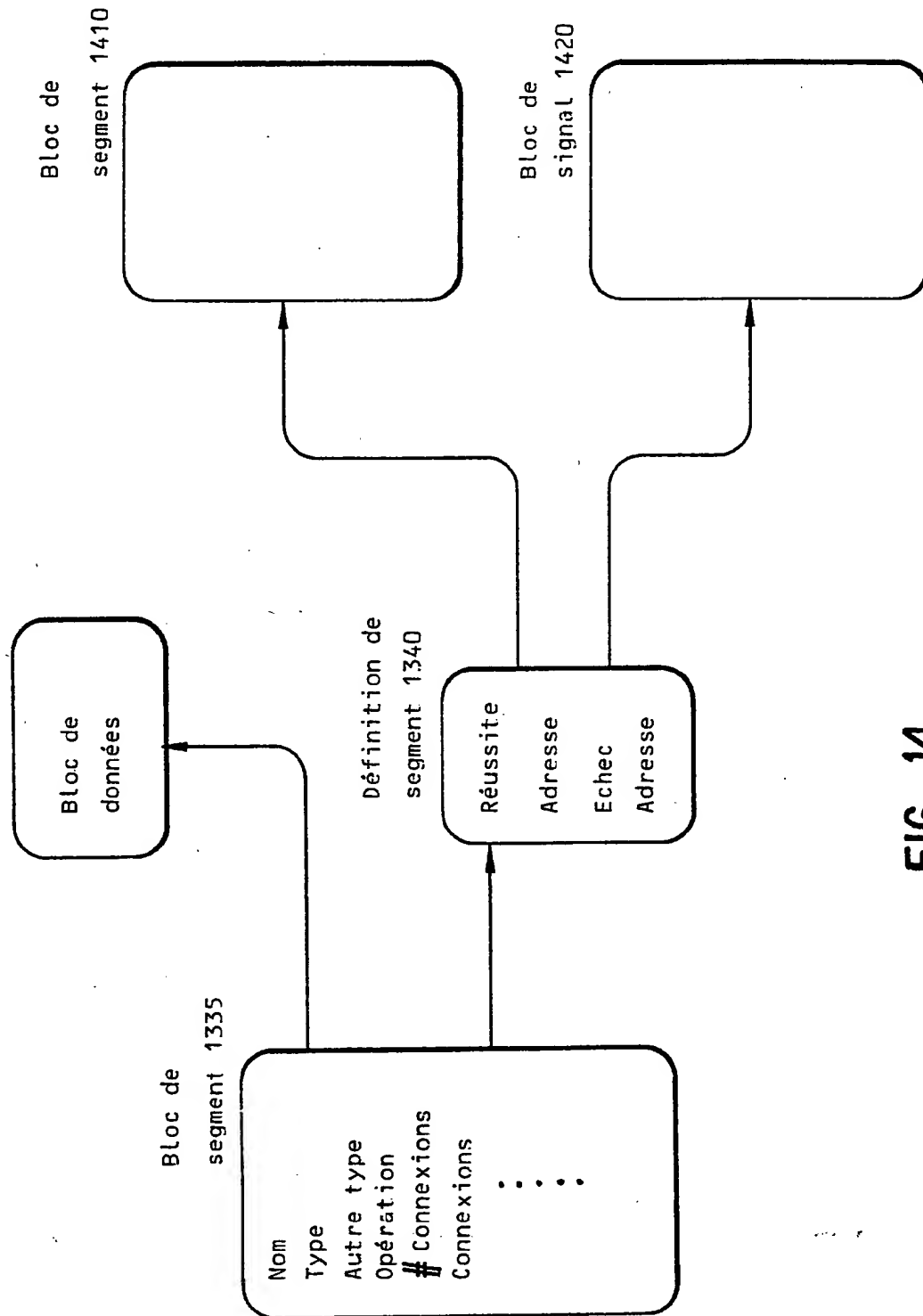


FIG. 14